

Contents

1	Routine/Function Prologues	2
1.0.1	parse_form_data (Source File: LIS_CFG_v1.1.cgi)	4
1.0.2	return_error (Source File: LIS_CFG_v1.1.cgi)	5
1.0.3	template (Source File: LIS_CFG_v1.1.cgi)	6

1 Routine/Function Prologues

This perl script allows users to create their own customized version of a configuration file, or "card file", for LIS. This file serves as the input to the LIS driver, which will run LIS using parameters that are set by the user via an HTML form and this script. This script writes the users' input to a template, which in turn writes the card file. The browser will then point to the newly created card file.

The first section of code processes the user's input. After the data is read in and parsed into files, the script does some simple error checking to make sure the lat/lon values for the domains exist. Once this is concluded, the script then writes a copy of the card file.

USES:

```
use strict;
```

INPUT PARAMETERS:

```
my ($stosend, $request_method, $query_string, %reg_data);
my ($key, $webmaster);
$webmaster = 'luther.lighty@nasa.gov';
```

```
$|=1; # disable buffering
```

```
#-----Read in the form data -----
$request_method = $ENV{'REQUEST_METHOD'};
```

```
if ($request_method eq "POST" ) {
    read (STDIN, $query_string, $ENV{'CONTENT_LENGTH'});
} else {
    &return_error (500, "Server Error",
                  "Server uses unsupported method");
}
```

```
# ----- Parse form data -----
&parse_form_data (\%reg_data, $query_string);
```

```
# ----- sanity processing of form data -----
# --- only [A-Za-Z0-9+-. \s] are allowed. -----
```

```
foreach $key (keys %reg_data) {
    $reg_data{$key} =~ tr#A-Za-z0-9\.\+\-\_\_\ / ##cd;
}
```

```
#-----Error check, Lat/Lon Ranges-----
#----Min lat/lon can't be greater than Max lat/lon
#----Execution Domain must be smaller or equal to the Parameter Domain
#
```

```
if ($reg_data{lonmin1} > $reg_data{lonmax1}) {
    &return_error (500, "Input Error",
```

```
        "Invalid Longitude Range!");
}

if ($reg_data{lonmin2} > $reg_data{lonmax2}) {
    &return_error (500, "Input Error",
        "Invalid Longitude Range!");
}

if ($reg_data{latmin1} > $reg_data{latmax1}) {
    &return_error (500, "Input Error",
        "Invalid Latitude Range!");
}

if ($reg_data{latmin2} > $reg_data{latmax2}) {
    &return_error (500, "Input Error",
        "Invalid Latitude Range!");
}

if ($reg_data{lonmin1} < $reg_data{lonmin2}) {
    &return_error (500, "Input Error",
        "Execution Domain must be smaller or equal to the Parameter Domain!");
}

if ($reg_data{lonmax1} < $reg_data{lonmax2}) {
    &return_error (500, "Input Error",
        "Execution Domain must be smaller or equal to the Parameter Domain!");
}

if ($reg_data{latmin1} < $reg_data{latmin2}) {
    &return_error (500, "Input Error",
        "Execution Domain must be smaller or equal to the Parameter Domain!");
}

if ($reg_data{latmax1} < $reg_data{latmax2}) {
    &return_error (500, "Input Error",
        "Execution Domain must be smaller or equal to the Parameter Domain!");
}

#----Error check, Default to global values if invalid-----
#-----Parameter Domain-----
if ($reg_data{lonmin1} < "-180") {
    $reg_data{lonmin1} = "-180";
}

if ($reg_data{lonmax1} > "180") {
    $reg_data{lonmax1} = "180";
}
}
```

```

if ($reg_data{latmin1} < "-60") {
    $reg_data{latmin1} = "-60";
}

if ($reg_data{latmax1} > "90") {
    $reg_data{latmax1} = "90";
}

#-----Execution Domain-----

if ($reg_data{lonmin2} < "-180") {
    $reg_data{lonmin2} = "-180";
}

if ($reg_data{lonmax2} > "180") {
    $reg_data{lonmax2} = "180";
}

if ($reg_data{latmin2} < "-60") {
    $reg_data{latmin2} = "-60";
}

if ($reg_data{latmax2} > "90") {
    $reg_data{latmax2} = "90";
}

#-----Write card file-----
open(INV, ">lis.crd") or die "can not open card file\n";
print INV template("card/template3.crd", \%reg_data);
close(INV);

#-----Feedback to the customer -----
print "Location: http://lis.gsfc.nasa.gov/lis.crd\n\n";
exit(0);

```

1.0.1 parse_form_data (Source File: LIS_CFG_v1.1.cgi)

Parse the form data and save it to files

```

sub parse_form_data
{
my ($FORM_DATA, $query_string) = @_;

my (@key_value_pairs, $key_value, $key, $value);

```

```

@key_value_pairs = split (/&/, $query_string);

foreach $key_value (@key_value_pairs) {
    ($key, $value) = split (/=/, $key_value);
    $value =~ s/\+/ /g;
    $value =~ s/%([\dA-Fa-f][\dA-Fa-f])/pack ("C", hex ($1))/eg;

    if (defined($$FORM_DATA{$key})) {
        $$FORM_DATA{$key} = join ("\0", $$FORM_DATA{$key}, $value);
    } else {
        $$FORM_DATA{$key} = $value;
    }
}
}
}

```

1.0.2 return_error (Source File: LIS_CFG_v1.1.cgi)

Return error message to user if an error occurs during processing

```

sub return_error
{
    my ($status, $keyword, $message) = @_;

    print "Content-type: text/html", "\n\n";
    print "Status: ", $status, " ", $keyword, "\n\n";

    print <<End_of_Error;

<HTML>
<HEAD>
    <TITLE> CGI Program - Unexpected Error </TITLE>
</HEAD>
<BODY>
<H1>$keyword</H1>
<HR>$message<HR>
Please contact
<a href="mailto: $webmaster">$webmaster</a> for more information.
</BODY>
</HTML>

End_of_Error

```

```
    exit(1);
}
```

1.0.3 template (Source File: LIS_CFG_v1.1.cgi)

Replaces quoted words with values in fillings hash

```
sub template {
    my ($filename, $fillings) = @_ ;
    my $text;
    local $/;
    local *F;
    open(F, "< $filename\0") || return;
    $text = <F>;
    close(F);
    #replace quoted workds with values in %$fillings hash
    $text =~ s{ %% ( .*? ) %% }
        { exists( $fillings->{$1} )
          ? $fillings->{$1}
          : ""
        }gsex;
    return $text;
}
1;      # return true of the module.
```